



# **NAVAL POSTGRADUATE SCHOOL**

**MONTEREY, CALIFORNIA**

## **THESIS**

**PERFORMANCE OF HYBRID EULERIAN-LAGRANGIAN  
SEMI-IMPLICIT TIME-INTEGRATORS FOR  
NONHYDROSTATIC MESOSCALE ATMOSPHERIC  
MODELING**

by

Thomas J. De Luca

September 2007

Thesis Advisors:

Francis X. Giraldo  
Rebecca E. Stone

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> September 2007	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> Performance of Hybrid Eulerian-Lagrangian Semi-Implicit Time-Integrators for Nonhydrostatic Mesoscale Atmospheric Modeling			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Thomas J. De Luca				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b> <p>In this thesis, the performance and accuracy of explicit, semi-implicit, and Hybrid Eulerian-Lagrangian Semi-Implicit (HELSE) time-integration methods used in atmospheric modeling are examined. Four test cases are analyzed: A density current, an inertial gravity wave, a rising thermal bubble, and a hydrostatic mountain wave. Strict attention is paid to computational time, stability criteria, and accuracy. The project aims to show increased efficiency using the HELSE method over fully semi-implicit methods, which, in turn, should be better than the split-explicit methods currently used in mesoscale models such as WRF, COAMPS, and the German LM model. This increase in efficiency allows for valuable computational resources to be used for other purposes, such as improved data assimilation, increased spatial resolution, or more detailed physics.</p>				
<b>14. SUBJECT TERMS</b> HELSE, semi-implicit, time-integration, atmospheric modeling, NWP			<b>15. NUMBER OF PAGES</b> 73	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**PERFORMANCE OF HYBRID  
EULERIAN-LAGRANGIAN SEMI-IMPLICIT TIME  
INTEGRATORS FOR NONHYDROSTATIC MESOSCALE  
ATMOSPHERIC MODELING**

Thomas J. De Luca  
Second Lieutenant, United States Air Force  
B.A., Cornell University, 2004  
B.S., Naval Postgraduate School, 2006

Submitted in partial fulfillment of the  
requirements for the degrees of

**MASTER OF SCIENCE IN METEOROLOGY  
MASTER OF SCIENCE IN APPLIED MATHEMATICS**

from the

**NAVAL POSTGRADUATE SCHOOL  
September 2007**

Author: Thomas J. De Luca

Approved by: Francis Giraldo, Co-Advisor

CDR Rebecca Stone, Co-Advisor

Philip Durkee, Chairman  
Department of Meteorology

Clyde Scandrett, Chairman  
Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

In this thesis, the performance and accuracy of explicit, semi-implicit, and Hybrid Eulerian-Lagrangian Semi-Implicit (HELSE) time-integration methods for use in atmospheric modeling are examined. Four test cases are analyzed: A density current, an inertial gravity wave, a rising thermal bubble, and a hydrostatic mountain wave. Strict attention is paid to computational time, stability criteria, and accuracy. The project aims to show increased efficiency using the HELSE method over semi-implicit methods, which, in turn, should be better than the split-explicit methods currently used in mesoscale models such as WRF, COAMPS, and the German LM model. This increase in efficiency allows for valuable computational resources to be used for other purposes, such as improved data assimilation, increased spatial resolution, or more detailed physics.

THIS PAGE INTENTIONALLY LEFT BLANK



# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION . . . . .</b>	<b>1</b>
<b>II.</b>	<b>BACKGROUND . . . . .</b>	<b>3</b>
A.	GOVERNING EQUATIONS . . . . .	3
1.	Equation Set . . . . .	3
B.	SPATIAL DISCRETIZATION . . . . .	3
C.	TIME-INTEGRATION . . . . .	6
1.	Eulerian Methods . . . . .	7
2.	Lagrangian Methods . . . . .	7
3.	Courant-Friedrichs-Levy Condition . . . . .	7
<b>III.</b>	<b>TIME-INTEGRATION METHODS . . . . .</b>	<b>9</b>
A.	EXPLICIT TIME-INTEGRATORS . . . . .	9
1.	Backwards Difference Formula . . . . .	9
2.	Leapfrog . . . . .	9
3.	Leapfrog Stability Analysis . . . . .	9
4.	Runge-Kutta Methods . . . . .	13
B.	SEMI-IMPLICIT TIME-INTEGRATORS . . . . .	14
C.	HYBRID EULERIAN-LAGRANGIAN SEMI-IMPLICIT TIME- INTEGRATORS . . . . .	18
1.	Operator-Integration-Factor Splitting Method . . . . .	19
<b>IV.</b>	<b>TEST CASES . . . . .</b>	<b>21</b>
A.	CASE 1: RISING THERMAL BUBBLE . . . . .	21
B.	CASE 2: LINEAR HYDROSTATIC MOUNTAIN WAVE . . . . .	21
C.	CASE 3: DENSITY CURRENT . . . . .	22
D.	CASE 4: INERTIA-GRAVITY WAVE . . . . .	23
<b>V.</b>	<b>RESULTS . . . . .</b>	<b>25</b>
A.	OVERVIEW . . . . .	25

B.	DETERMINING THE MAXIMUM USABLE TIME STEP . . .	25
C.	CASE 1: RISING THERMAL BUBBLE . . . . .	27
1.	Courant Number and Computational Cost . . . . .	27
2.	Accuracy . . . . .	28
3.	Comparison and Conclusions . . . . .	30
D.	CASE 2: LINEAR HYDROSTATIC MOUNTAIN WAVE . . . .	31
1.	Courant Number and Computational Cost . . . . .	31
2.	Accuracy . . . . .	32
3.	Comparison and Conclusions . . . . .	34
E.	CASE 3: DENSITY CURRENT . . . . .	35
1.	Courant Number and Computational Cost . . . . .	35
2.	Accuracy . . . . .	36
3.	Comparison and Conclusions . . . . .	39
F.	CASE 4: INERTIAL GRAVITY WAVE . . . . .	39
1.	Courant Number and Computational Cost . . . . .	39
2.	Accuracy . . . . .	40
3.	Comparison and Conclusions . . . . .	43
<b>VI.</b>	<b>CONCLUSIONS AND RECOMMENDATIONS . . . . .</b>	<b>45</b>
	<b>APPENDIX A. COEFFICIENTS FOR RK35 METHOD . . . . .</b>	<b>47</b>
	<b>APPENDIX B. STABILITY ANALYSIS OF LEAPFROG METHODS</b>	
	<b>WITH TIME-FILTERING . . . . .</b>	<b>49</b>
1.	EXPLICIT LEAPFROG WITH TIME-FILTER . . . . .	49
2.	SEMI-IMPLICIT LEAPFROG WITH TIME-FILTER . . . . .	51
	<b>APPENDIX C. COMPLETE RESULTS FOR HELSI METHODS .</b>	<b>53</b>
	<b>LIST OF REFERENCES . . . . .</b>	<b>55</b>
	<b>INITIAL DISTRIBUTION LIST . . . . .</b>	<b>57</b>

# LIST OF FIGURES

1.	The stability of the explicit leapfrog time-integrator. Figure a) has no time-filter, while figure b) has a time-filter weight of $\epsilon=.05$ . The solid lines represent the physical solutions while the dashed lines represent the computational modes. . . . .	13
2.	Stability of the semi-implicit leapfrog time integrator with no time-filter. Figure a) uses a value of $v = .5$ and Figure b) uses a value of $v = .6$ . . . . .	17
3.	Stability of the semi-implicit leapfrog time integrator with a time filter weight of $\epsilon = .05$ . Figure a) uses a value of $v = .5$ and Figure b) uses a value of $v = .6$ . . . . .	17
4.	A schematic of the OIFS trajectory computation. The curved line is the actual trajectory, while the arrows denote the paths followed to compute the departure points using an RK2 approximation. Each arrow represents a two-stage process. . . . .	20
5.	An example of the HELSI time-integrator retaining stability but not accuracy. In figure a), case 4 is time-integrated with a timestep of 4.75 seconds using the HELSI method, a RK2 OIFS method, and one substep. In figure b) case 4 is time integrated using a timestep of 5.75 seconds using the same HELSI method. While the integration remains stable, the result is obviously unusable. . . . .	26
6.	Case1: A comparison of potential temperature perturbations using a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators for case 1 at 3000 sec. . . . .	30
7.	Case 1: A view of potential temperature perturbations using the HELSI time-integrator with RK4 depart method and 4 substeps at 3000 sec. .	30

8.	Case 2: A comparison of vertical velocity perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 1 hr. . . . .	34
9.	Case 2: A comparison of the vertical velocity perturbations for a) the HELSI method with depart method RK2 and 1 substep and b) the HELSI method with depart method RK3 and 1 substep at 1 hr. . . . .	34
10.	Case 3: A comparison of potential temperature perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 900 sec. . . . .	38
11.	Case 3: Potential temperature perturbations for the HELSI method with depart method RK3 and 1 substep at 900 sec. . . . .	38
12.	Case 4: A comparison of potential temperature perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 3000 sec. . . . .	42
13.	Case 4: A comparison of potential temperature perturbations for the a) HELSI method with depart method RK2 and 1 substep and b) the HELSI method with depart method RK2 and 3 substeps at 300 sec. . . . .	43

# LIST OF TABLES

I.	Case 1: Courant number and computational cost of explicit methods. . .	27
II.	Case 1: Courant number and computational cost of semi-implicit methods. .	27
III.	Case 1: Courant number and computational cost of HELSI method. . .	28
IV.	Case 1: A comparison of potential temperature perturbations for the explicit time-integrators. . . . .	28
V.	Case 1: A comparison of potential temperature perturbations for the semi-implicit time-integrators. . . . .	29
VI.	Case 1: Potential temperature perturbations for the HELSI time-integrator. .	29
VII.	Case 2: Courant number and computational cost of explicit methods. . .	31
VIII.	Case 2: Courant number and computational cost of semi-implicit methods. .	31
IX.	Case 2: Courant number and computational cost of HELSI methods. . .	32
X.	Case 2: A comparison of RMS errors for the explicit time-integrators. . .	33
XI.	Case 2: A comparison of RMS errors for the semi-implicit time-integrators. .	33
XII.	Case 2: A comparison of RMS errors for the HELSI time-integrators. . .	33
XIII.	Case 3: Courant number and computational cost of explicit methods. . .	35
XIV.	Case 3: Courant number and computational cost of semi-implicit methods. .	36
XV.	Case 3: Courant number and computational cost of HELSI methods. . .	36
XVI.	Case 3: A comparison of potential temperature perturbations for the explicit time-integrators. . . . .	37
XVII.	Case 3: A comparison of potential temperature perturbations for the semi-implicit time-integrators. . . . .	37
XVIII.	Case 3: Potential temperature perturbations for the HELSI time-integrator. .	38
XIX.	Case 4: Courant number and computational cost of explicit methods. . .	39
XX.	Case 4: Courant number and computational cost of semi-implicit methods. .	40
XXI.	Case 4: Courant number and computational cost of HELSI methods. . .	40

XXII. Case 4: A comparison of potential temperature perturbations for the explicit time-integrators. . . . .	41
XXIII. Case 4: A comparison of potential temperature perturbations for the semi-implicit time-integrators. . . . .	41
XXIV. Case 4: A comparison of potential temperature perturbations for the HELSI time-integrators. . . . .	42
XXV. Case 1: Courant number and computational cost of HELSI methods. .	53
XXVI. Case 2: Courant number and computational cost of HELSI methods. .	53
XXVII. Case 3: Courant number and computational cost of HELSI methods. .	54
XXVIII. Case 4: Courant number and computational cost of HELSI methods. .	54

# ACKNOWLEDGMENTS

I would like to thank my advisors, Francis X. Giraldo and CDR Rebecca Stone, without whom this thesis would not have been possible. I would also like to thank the Naval Postgraduate School, the Meteorology and Mathematics departments at NPS, The Air Force Institute of Technology, The United States Air Force, and The United States Navy.

THIS PAGE INTENTIONALLY LEFT BLANK



# I. INTRODUCTION

A numerical weather prediction (NWP) modeling system is a method of predicting the future state of the atmosphere using data collected about the current state of the atmosphere and known information about atmospheric behavior. These behavioral properties are approximated using a mathematical model, called the governing equations, and by physical parametrizations.

An operational NWP modeling system is a multi-component system. Processes for accurate data assimilation, forecast integration, and physical parametrizations are vital to produce an accurate forecast. Improving any one of the components should increase the overall accuracy of the NWP system, received as “NWP guidance” by forecasters, and should thus help produce a better operational forecast.

Improvements to any component, however, generally require increased computational expense. When improvements to NWP systems are made, timeliness must also be maintained. Increasing computational requirements must be matched by increased computational capacity, or increased computational efficiency, or both. Limited budgetary resources demand that significant attention ought to be placed on the computational efficiency approach. One area where a significant improvement in efficiency may be achieved is the numerical time-integration methods used to march the governing equations forward in time, and it is the purpose of this thesis to advance the prospect of implementing more efficient methods in Department of Defense NWP systems.

Many current non-hydrostatic mesoscale NWP models such as the Weather Research and Forecasting (WRF) model and The U.S. Navy’s Coupled Ocean Atmosphere Mesoscale Prediction System (COAMPS) use split-explicit time integration methods. Semi-implicit methods suggested by Giraldo [1] offer a significant improvement in efficiency. Hybrid Eulerian-Lagrangian Semi-Implicit (HELSI) time-integrators may offer even further improvements.

Here, the accuracy and efficiency of explicit, semi-implicit, and HELSI methods are explored. Four test cases are examined: a rising thermal bubble, a linear hydrostatic mountain wave, a density current, and an inertia-gravity wave. The various time integrators are used to calculate the perturbations in Exner pressure, velocity, and potential temperature for each of these test cases. Careful attention is paid to wallclock computational time, stability criteria, and accuracy.

## II. BACKGROUND

### A. GOVERNING EQUATIONS

For this study, equation set 1 of Giraldo and Restelli [2] is used. This equation set is the non-conservative form of the Euler equations. It is the two-dimensional version of the equations currently used in operational NWP models such as COAMPS and the German LM model.

#### 1. Equation Set

$$\begin{aligned}\frac{\partial \pi}{\partial t} + \mathbf{u} \cdot \nabla \pi + \frac{R}{c_v} \pi \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + c_p \theta \nabla \pi &= -g \mathbf{k} + \mu \nabla^2 \mathbf{u} \\ \frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta &= \mu \nabla^2 \theta\end{aligned}\tag{2.1}$$

Here,  $\pi = \left(\frac{P}{P_0}\right)^{R/c_p}$  is the Exner pressure,  $\mathbf{u} = (u, w)^T$  is the 2-dimensional velocity field, and  $\theta = \frac{T}{\pi}$  is the potential temperature. The solution vector is  $(\pi, \mathbf{u}^T, \theta)^T$ .

### B. SPATIAL DISCRETIZATION

A finite element spatial discretization on quadrilateral elements, as defined by Giraldo and Restelli [2], is presented. This decomposition of the global domain into a multitude of smaller domains is designed to fully exploit the multiple processor architecture currently used in high performance computing [1]. In order to solve the equations numerically, they must first be recast in the form

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{S}(\mathbf{q})\tag{2.2}$$

where  $\mathbf{S}(\mathbf{q})$  contains the source terms of the governing equations. Then, the global domain,  $\Omega$ , is broken up into  $N_e$  non-overlapping quadrilateral elements such that

$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e.$$

In order to perform calculus operations, a non-singular mapping  $\mathbf{x} = \Psi(\boldsymbol{\xi})$  from the physical Cartesian coordinate system  $\mathbf{x} = (x, z) \in \Omega_e$  to the reference coordinate system  $\boldsymbol{\xi} = (\xi, \eta)$  which is defined in each element such that  $(\xi, \eta) \in [-1, +1]^2$  in each element. Also used is the transformation Jacobian,  $J = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ , which is associated with the local mapping,  $\boldsymbol{\xi}$ . This mapping is then used to define the local representation of  $\mathbf{q}$ , the solution vector, and the approximation of the calculus operations.

The structure of the reference element,  $\mathbf{l}$ , which is spanned by  $\boldsymbol{\xi} \in [-1, 1]^2$ , makes a natural representation of the local element-wise solution  $\mathbf{q}$  by an  $N$ th order polynomial as follows:

$$\mathbf{q}_N(\mathbf{x}) = \sum_{k=1}^K \psi_k(\mathbf{x}) \mathbf{q}_N(\mathbf{x}_k)$$

where  $\mathbf{x}_k$  represents  $K = (N + 1)^2$  grid points and  $\psi_k(\mathbf{x})$  reflects the associated multivariate Lagrange polynomials; these polynomials are the basis functions used in standard finite elements. The square structure of the reference element allows the representation of the Lagrange polynomial by a tensor-product

$$\psi_k(\mathbf{x}) = h_i(\xi(\mathbf{x}))h_j(\eta(\mathbf{x})), \quad (2.3)$$

where  $i, j = 0, \dots, N$ .

For the grid points  $(\xi_i, \eta_j)$  the Legendre-Gauss-Lobatto (LGL) points are chosen, given as the tensor-product of the roots of

$$(1 - \xi^2)P'_N(\xi) = 0,$$

where  $P_N(\xi)$  is the  $N$ th order Legendre polynomial. This will simplify the algorithm, as the LGL points will be used elsewhere.

The one-dimensional Lagrange polynomials,  $h_i(\xi)$  are

$$h_i(\xi) = -\frac{1}{N(N+1)} \frac{(1 - \xi^2)P'_N(\xi)}{(\xi - \xi_i)P'_N(\xi_i)},$$

and likewise for  $h_j(\eta)$  [3].

Having chosen the LGL points makes the approximation of local element integrals straightforward

$$\int_{\Omega_e} q(\mathbf{x}) d\mathbf{x} = \int_{\mathbf{l}} q(\boldsymbol{\xi}) J(\boldsymbol{\xi}) d\boldsymbol{\xi} \simeq \sum_{i,j=0}^N \omega(\xi_i) \omega(\eta_j) q(\xi_i, \eta_j) |J(\xi_i, \eta_j)|,$$

where  $J$  represents the local Jacobian of the transformation between  $\Omega_e$  and  $\mathbf{l}$ , and  $\omega(\xi_i)$  and  $\omega(\eta_j)$  are the Gaussian quadrature weights,

$$\omega(\xi_i) = \frac{2}{N(N+1)} \left( \frac{1}{P_N(\xi_i)} \right)^2,$$

associated with the one-dimensional LGL quadrature.

In the spectral element (SE) method, which is a high-order finite element method, a polynomial expansion

$$\mathbf{q}_N(\mathbf{x}, t) = \sum_{k=1}^K \psi_k(\mathbf{x}) \mathbf{q}_k(t) \quad (2.4)$$

is used to approximate  $\mathbf{q}$ . The variational statement of equation 2.2 is: find  $\mathbf{q} \in H^1(\Omega) \forall \psi \in H^1$  such that

$$\int_{\Omega} \psi \left[ \frac{\partial \mathbf{q}_N}{\partial t} - S(\mathbf{q}_N) \right] d\Omega = 0 \quad (2.5)$$

where  $H^1(\Omega)$  is defined as the space of all  $C^0$  continuous functions with functions and first derivatives belonging to  $L^2(\Omega)$ , the square integrable functions. i.e.

$$\int_{\Omega} |\mathbf{q}|^2 d\Omega < \infty, \forall \mathbf{q} \in \Omega.$$

Simplifying Eq. (2.5) by virtue of the SE discretization yields the global matrix problem

$$\begin{aligned} \frac{\partial \pi_I}{\partial t} + \mathbf{u}_I^T M_I^{-1} \mathbf{D}_{IJ} \pi_J + \frac{R}{c_v} \pi_I M_I^{-1} \mathbf{D}_{IJ}^T \mathbf{u}_J &= 0 \\ \frac{\partial \mathbf{u}_I}{\partial t} + \mathbf{u}_I^T M_I^{-1} \mathbf{D}_{IJ} \mathbf{u}_J + c_p \theta_I M_I^{-1} \mathbf{D}_{IJ} \pi_J &= -g\mathbf{k} - \mu L_{IJ} \mathbf{u}_J \\ \frac{\partial \theta_I}{\partial t} + \mathbf{u}_I^T M_I^{-1} \mathbf{D}_{IJ} \theta_J &= -\mu L_{IJ} \theta_J \end{aligned} \quad (2.6)$$

where

$$M = \bigwedge_{e=1}^{N_e} M^e = \bigwedge_{e=1}^{N_e} \int_{\Omega_e} \psi_i \psi_j d\Omega_e,$$

is the global mass matrix

$$\mathbf{D} = \bigwedge_{e=1}^{N_e} \mathbf{D}^e = \bigwedge_{e=1}^{N_e} \int_{\Omega_e} \psi_i \nabla \psi_j d\Omega_e,$$

is the global differentiation matrix and

$$L = \bigwedge_{e=1}^{N_e} L^e = \bigwedge_{e=1}^{N_e} \int_{\Omega_e} \nabla \psi_i \nabla \psi_j d\Omega_e$$

is the global Laplacian matrix. In the SE method, the local element-wise matrices are built first,

$$M_{ij}^e = w_i^e |J_i^e| \delta_{ij}, \quad (2.7)$$

$$\mathbf{D}_{ij}^e = w_i^e |J_i^e| \nabla \psi_j(\mathbf{x}_i), \quad (2.8)$$

$$L_{ij}^e = \sum_{i=1}^K w_i^e |J^e| \nabla \psi_i(\mathbf{x}_i) \cdot \nabla \psi_j(\mathbf{x}_i), \quad (2.9)$$

where  $w^e$  are the quadrature weights,  $J^e$  is the Jacobian,  $i = 1, \dots, K$  are the local element grid points,  $e = 1, \dots, N_e$  are the elements covering the global domain, and  $I, J = 1, \dots, N_p$  are the global grid points. Finally, the direct stiffness summation (DSS) operator

$$\bigwedge_{e=1}^{N_e}$$

is used to map the local element grid points  $i = 1, \dots, K$  and elements  $e = 1, \dots, N_e$  to the corresponding global grid points via the mapping  $(i, e) \longrightarrow (I)$  where  $I = 1, \dots, N_p$  are the global grid points.

## C. TIME-INTEGRATION

A numerical time-integrator is a numerical method used to approximate the time-derivative of the equation

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{S}(\mathbf{q}). \quad (2.10)$$

Here,  $\mathbf{q} = (\pi, \mathbf{u}^T, \theta)^T$  is the solution vector of Equation 2.1, and  $\mathbf{S}(\mathbf{q})$  is the spatial discretization of the right hand side of Equation 2.1.

There are two basic types of time-integrators: Eulerian and Lagrangian.[4]

## 1. Eulerian Methods

Eulerian methods use a fixed frame of reference. As an example, equation 2.10 is written in Eulerian form. As another example, the advection equation in Eulerian form is

$$\frac{\partial \mathbf{q}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{q} \quad (2.11)$$

where  $\mathbf{u}$  is the velocity vector

## 2. Lagrangian Methods

In contrast, Lagrangian methods use a moving frame of reference. Therefore, equation 2.10 written in Lagrangian form is

$$\frac{d\mathbf{q}}{dt} = \mathbf{S}_L(\mathbf{q}) \quad (2.12)$$

where

$$\frac{d}{dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla. \quad (2.13)$$

and

$$\mathbf{u} = \frac{d\mathbf{x}}{dt}. \quad (2.14)$$

The advection equation written in Lagrangian form can be simplified to

$$\frac{d\mathbf{q}}{dt} = 0. \quad (2.15)$$

## 3. Courant-Friedrichs-Levy Condition

A vital component to a discussion on time-integrators is the Courant number, which measures the amount of information traversing a grid cell ( $\Delta x$ ) in a given timestep ( $\Delta t$ ).

For example, the one-dimensional Courant number is defined to be

$$C \equiv \frac{u\Delta t}{\Delta x}. \quad (2.16)$$

The Courant-Friedrichs-Levy (CFL) condition is the largest Courant number that a time-integrator can use. In general, increasing the Courant number is a central goal to increasing the overall efficiency of the time-integration method.



### III. TIME-INTEGRATION METHODS

#### A. EXPLICIT TIME-INTEGRATORS

Explicit Time-Integrators are, by far, the simplest of the methods examined. They have been proven to be very accurate. They are, however, quite inefficient. One particularly nice feature is that only knowledge of the solution at previous time steps is needed. In general, an explicit time-integrator is of the form

$$\mathbf{q}^{n+1} = \sum_{i=0}^{M_E} \alpha_i \mathbf{q}^{n-i} + \gamma \Delta t \sum_{i=0}^{M_E} \beta_i \mathbf{S}(\mathbf{q}^{n-i}). \quad (3.1)$$

##### 1. Backwards Difference Formula

The second order backwards difference formula (BDF2) is

$$\mathbf{q}^{n+1} = \frac{4}{3}\mathbf{q}^n + \frac{1}{3}\mathbf{q}^{n-1} + \frac{2}{3}\Delta t(2\mathbf{S}(\mathbf{q}^n) + \mathbf{S}(\mathbf{q}^{n-1})). \quad (3.2)$$

##### 2. Leapfrog

The second order leapfrog formula (LF2) is

$$\mathbf{q}^{n+1} = \mathbf{q}^{n-1} + 2\Delta t(\mathbf{S}(\mathbf{q}^n)). \quad (3.3)$$

##### 3. Leapfrog Stability Analysis

Due to the existence of Leapfrog's computational mode, a Robert-Asselin time-filter is necessary to preserve stability. At each timestep, the following filter is applied

$$\tilde{\mathbf{q}}^n = \mathbf{q}^n + \epsilon(\mathbf{q}^{n+1} - 2\mathbf{q}^n + \tilde{\mathbf{q}}^{n-1}) \quad (3.4)$$

where  $\tilde{\mathbf{q}}$  is the time-filtered variable and  $\epsilon$  is the time-filter weight. A brief linear stability analysis of a simple equation demonstrates this necessity.

##### *a. No Time-filter*

Consider the equation

$$\frac{\partial \mathbf{q}}{\partial t} = ik\mathbf{q} \quad (3.5)$$

whose solution is known to be

$$\mathbf{q} = Ae^{ikt} \quad (3.6)$$

where A is any constant. If the equations are discretized, then, at timestep n,  $t = n\Delta t$ ,

$$\mathbf{q}^n = Ae^{ikn\Delta t} = A(e^{ik\Delta t})^n \quad (3.7)$$

and

$$\mathbf{S}(\mathbf{q})^n = ik\mathbf{q}^n = ikA(e^{ik\Delta t})^n. \quad (3.8)$$

Consider the case where there is no time-filter, ie  $\epsilon = 0$ , then

$$A(e^{ik\Delta t})^{n+1} = A(e^{ik\Delta t})^{n-1} + 2\Delta t ikA(e^{ik\Delta t})^n \quad (3.9)$$

or

$$\lambda^{n+1} = \lambda^{n-1} + 2ik\Delta t\lambda^n \quad (3.10)$$

where  $\lambda = e^{ik\Delta t}$  is the amplification factor. If  $|\lambda| > 1$ , the solution grows exponentially fast, and, thus, is unstable. Simplifying yields

$$\lambda^2 - 2ik\Delta t\lambda - 1 = 0 \quad (3.11)$$

which has the solution

$$\lambda = ik\Delta t \pm \sqrt{-(k\Delta t)^2 + 1} \quad (3.12)$$

Note that the amplification factor can take on two values:  $\lambda = ik\Delta t + \sqrt{-(k\Delta t)^2 + 1}$ , which is the physical solution, while  $\lambda = ik\Delta t - \sqrt{-(k\Delta t)^2 + 1}$  is called the computational mode. In order to examine  $|\lambda|$ , two cases are considered:  $|k\Delta t| > 1$  and  $|k\Delta t| \leq 1$

If  $|k\Delta t| > 1$  the amplification factor is purely imaginary. Thus, the magnitude of the two modes are different. The magnitude of the amplification factor of the physical solution is  $\sqrt{2(k\Delta t)^2 + 2k\Delta t\sqrt{(k\Delta t)^2 - 1}} - 1$ , which is always greater than one. Therefore, the method is always unstable here. For completeness, the magnitude of the amplitude factor of the computational mode is  $\sqrt{2(k\Delta t)^2 - 2k\Delta t\sqrt{(k\Delta t)^2 - 1}} - 1$ .

When  $|k\Delta t| \leq 1$ , the term inside the square root is real while the other term is imaginary, and the magnitude of the amplification factor for both the physical solution and computational mode is

$$(k\Delta t)^2 - (k\Delta t)^2 + 1 = 1 \quad (3.13)$$

which is neutrally stable.

Since the computational mode is as large as the physical solution, non-linear interactions can affect the physical solution and cause instabilities [4]. This problem is corrected with time-filtering. It is necessary to note, however, that this will reduce the accuracy of leapfrog to first order.

### ***b. Time-filtered***

Now consider the case where the time-filter is applied. Starting with equation 3.3 and 3.4, the definition of the time-filtered leapfrog method becomes

$$\mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n-1} + 2\Delta t(\mathbf{S}(\mathbf{q}^n)). \quad (3.14)$$

Using the same equation

$$\frac{\partial \mathbf{q}}{\partial t} = ik\mathbf{q} \quad (3.15)$$

gives

$$\mathbf{q} = Ae^{ikt} \quad (3.16)$$

for the non-filtered solution and

$$\tilde{\mathbf{q}} = \tilde{A}e^{ikt} \quad (3.17)$$

for the time-filtered solution. Substituting  $\lambda = e^{ikt}$ , and dividing by  $\lambda^{n-1}$ , the time-filtered leapfrog method becomes

$$A\lambda^2 = \tilde{A} + 2Aik\Delta t\lambda \quad (3.18)$$

where

$$\tilde{A} = A + \epsilon(A\lambda - 2A + \tilde{A}\lambda^{-1}) \quad (3.19)$$

which follows directly from the definition of the time-filter. Simplifying and solving using the quadratic equation yields

$$\lambda = \epsilon + \Delta tik \pm \sqrt{(\epsilon + \Delta tik)^2 + 1 - 2\epsilon - 2\Delta tik\epsilon}. \quad (3.20)$$

For a more rigorous treatment, see Appendix B. The magnitude of both the physical solution and computational mode are plotted using a value of  $\epsilon = .05$ . Clearly, the computational mode is damped, and, thus, the problems of nonlinear interactions have been addressed.

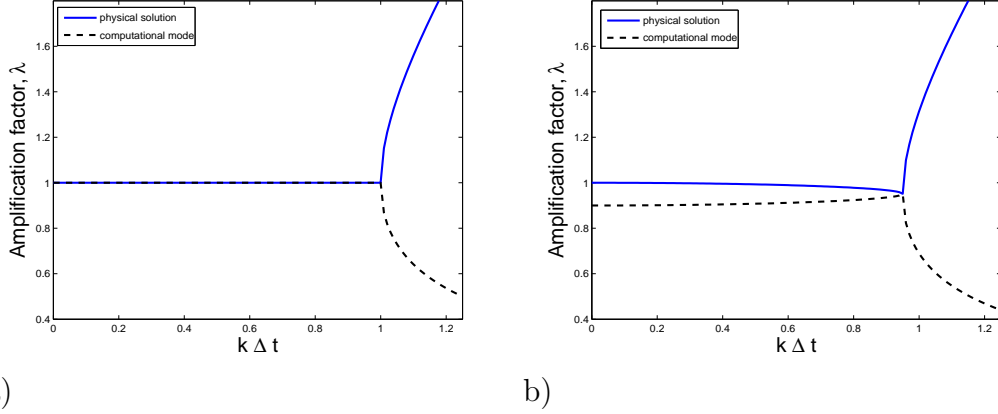


Figure 1. The stability of the explicit leapfrog time-integrator. Figure a) has no time-filter, while figure b) has a time-filter weight of  $\epsilon=.05$ . The solid lines represent the physical solutions while the dashed lines represent the computational modes.

#### 4. Runge-Kutta Methods

The Runge-Kutta (R-K) methods examined are part of a class of methods known as strongly stability preserving (SSP) methods proposed by Cockburn and Shu [5] and Ruuth and Spiteri [6]. They are third order, multi-stage methods. These methods offer larger stability regions than the previous explicit methods. The trade off, however, is that more computations are required per timestep. The first of these methods is the three-stage third order R-K method (RK3)

$$\begin{aligned}
 \mathbf{q}^{(1)} &= \mathbf{q}^n + \Delta t \mathbf{S}(\mathbf{q}^n) \\
 \mathbf{q}^{(2)} &= \frac{3}{4} \mathbf{q}^n + \frac{1}{4} \mathbf{q}^{(1)} + \frac{1}{4} \Delta t \mathbf{S}(\mathbf{q}^{(1)}) \\
 \mathbf{q}^{(n+1)} &= \frac{1}{3} \mathbf{q}^n + \frac{2}{3} \mathbf{q}^{(2)} + \frac{2}{3} \Delta t \mathbf{S}(\mathbf{q}^{(2)}).
 \end{aligned} \tag{3.21}$$

The second is the four-stage third order R-K method (RK34)

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + \frac{1}{2}\Delta t \mathbf{S}(\mathbf{q}^n) \\
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + \frac{1}{2}\Delta t \mathbf{S}(\mathbf{q}^{(1)}) \\
\mathbf{q}^{(3)} &= \frac{2}{3}\mathbf{q}^n + \frac{1}{3}\mathbf{q}^{(2)} + \frac{1}{6}\Delta t \mathbf{S}(\mathbf{q}^{(2)}) \\
\mathbf{q}^{(n+1)} &= \mathbf{q}^{(3)} + \frac{1}{2}\Delta t \mathbf{S}(\mathbf{q}^{(3)}).
\end{aligned} \tag{3.22}$$

The final is the five-stage third order R-K method (RK35)

$$\begin{aligned}
\mathbf{q}^{(1)} &= \mathbf{q}^n + a_1\Delta t \mathbf{S}(\mathbf{q}^n) \\
\mathbf{q}^{(2)} &= \mathbf{q}^{(1)} + a_2\Delta t \mathbf{S}(\mathbf{q}^{(1)}) \\
\mathbf{q}^{(3)} &= a_3\mathbf{q}^n + a_4\mathbf{q}^{(2)} + a_5\Delta t \mathbf{S}(\mathbf{q}^{(2)}) \\
\mathbf{q}^{(3)} &= a_6\mathbf{q}^n + a_7\mathbf{q}^{(3)} + a_8\Delta t \mathbf{S}(\mathbf{q}^{(3)}) \\
\mathbf{q}^{(n+1)} &= a_9\mathbf{q}^{(4)} + a_{10}\mathbf{q}^{(2)} + a_{11}\Delta t \mathbf{S}(\mathbf{q}^{(4)})
\end{aligned} \tag{3.23}$$

whose coefficients  $a_i$  are listed in Appendix A.

## B. SEMI-IMPLICIT TIME-INTEGRATORS

Implicit time-integrators are more complicated than explicit integrators because they require knowledge of the solution at the current timestep as well as the previous ones. The benefit is that they allow for a higher Courant number, especially important in the calculation of fast propagating waves. In general, an implicit time-integrator is of the form

$$\mathbf{q}^{n+1} = \sum_{i=0}^{M_I} \alpha_i \mathbf{q}^{n-i} + \gamma \Delta t \sum_{i=-1}^{M_I} \beta_i \mathbf{S}(\mathbf{q}^{n-i}) \tag{3.24}$$

The semi-implicit (SI) method treats some terms implicitly and others explicitly. The SI method of Giraldo is presented [1].

A SI time-integrator splits the source terms into linear and nonlinear terms

$$\frac{\partial \mathbf{q}}{\partial t} = \{\mathbf{S}(\mathbf{q}) - \delta \mathbf{L}(\mathbf{q})\} + [\delta \mathbf{L}(\mathbf{q})] \quad (3.25)$$

where the terms in braces are time-integrated explicitly, while the terms in the square brackets are time-integrated implicitly.  $\mathbf{L}$  represents the linearization of  $\mathbf{S}$ . Therefore, nonlinear terms are integrated explicitly and the linear terms implicitly. Finally,  $\delta = 0$  for a fully explicit method and  $\delta = 1$  for a semi-implicit method. For this study, two semi-explicit methods are examined: a second order semi-implicit backwards difference formula (BDF2 SI)

$$\mathbf{q}^{n+1} = \frac{4}{3}\mathbf{q}^n - \frac{1}{3}\mathbf{q}^{n-1} + \frac{2}{3}\Delta t(2\mathbf{S}(\mathbf{q})^n - \mathbf{S}(\mathbf{q}^{n-1})) + \delta \frac{2}{3}\Delta t \mathbf{L}(\mathbf{q}^{n+1} - 2\mathbf{q}^n + \mathbf{q}^{n-1}) \quad (3.26)$$

and a second order semi-implicit leapfrog method (LF2 SI)

$$\mathbf{q}^{n+1} = \mathbf{q}^{n-1} + 2\Delta t \mathbf{S}(\mathbf{q})^n + \delta 2\Delta t \mathbf{L}(v\mathbf{q}^{n+1} - \mathbf{q}^n + (1-v)\mathbf{q}^{n-1}). \quad (3.27)$$

For this project, a value of  $v = 0.6$  was chosen. This is a common value for atmospheric models [7]. To illustrate the reasoning, a linear stability analysis of the semi-implicit leapfrog method is performed. Again, consider the equation

$$\frac{\partial \mathbf{q}}{\partial t} = ik\mathbf{q}$$

with solution

$$\mathbf{q} = Ae^{ikt}.$$

Since this is a linear equation,  $\mathbf{L}(\mathbf{q}) = \mathbf{S}(\mathbf{q})$ , and the semi-implicit leapfrog method is reduced to

$$\mathbf{q}^{n+1} = \mathbf{q}^{n-1} + 2\Delta t[\mathbf{L}(v\mathbf{q}^{n+1}) + \mathbf{L}((1-v)\mathbf{q}^{n-1})]. \quad (3.28)$$

Considering the case where  $v = .5$ , and again letting  $\lambda = e^{ik\Delta t}$ , the equation reduces to

$$\lambda^{n+1} = \lambda^{n-1} + \Delta t(ik\lambda^{n+1} + ik\lambda^{n-1})$$

which has a solution

$$\lambda = \pm \sqrt{\frac{1 + ik\Delta t}{1 - ik\Delta t}} \quad (3.29)$$

where the positive value represents the physical solution and the negative value represents the computational mode. For both modes,  $|\lambda| = 1$ , meaning the solution is neutrally stable. Again, as in the case of explicit leapfrog without time-filtering, nonlinear interactions can affect the physical solution and cause instabilities [4].

Now, consider the case where  $v = .6$ . The semi-implicit leapfrog method becomes

$$\mathbf{q}^{n+1} = \mathbf{q}^{n-1} + 2\Delta t[\mathbf{L}(.6\mathbf{q}^{n+1}) + \mathbf{L}(.4\mathbf{q}^{n-1})]. \quad (3.30)$$

Again, letting  $\lambda = e^{ik\Delta t}$ , the method is reduced to

$$\lambda^{n+1} = \lambda^{n-1} + 2\Delta t(.6ik\lambda^{n+1} + .4ik\lambda^{n-1})$$

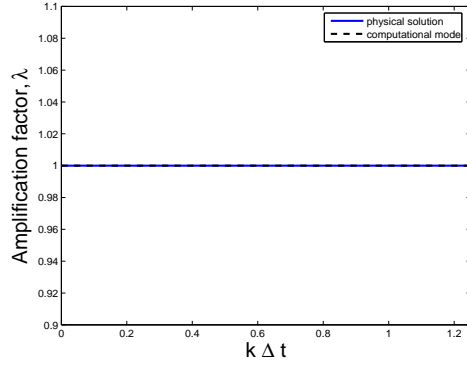
which has solution

$$\lambda = \pm \sqrt{\frac{1 + 0.8ik\Delta t}{1 - 1.2ik\Delta t}}. \quad (3.31)$$

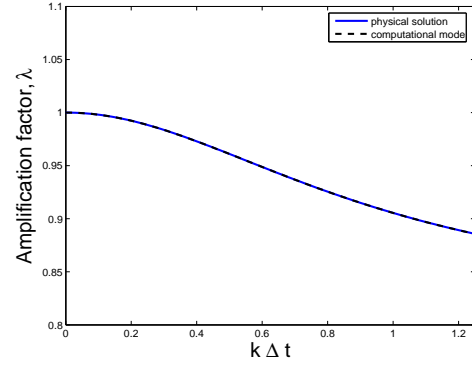
Plotting  $|\lambda_+|$  and  $|\lambda_-|$ , shows that the amplification factor has been damped, and the problems associated with the nonlinear interactions have been addressed.

Still, the computational mode is as large as the physical solution. Again, time-filtering corrects this problem. For details, see Appendix B.



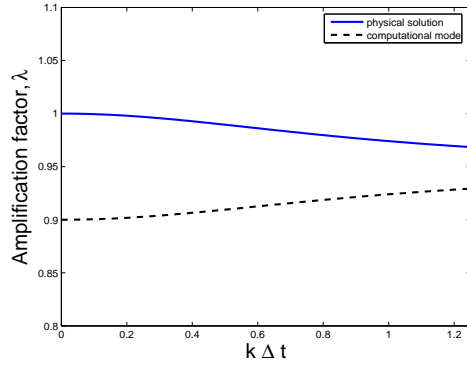


a)

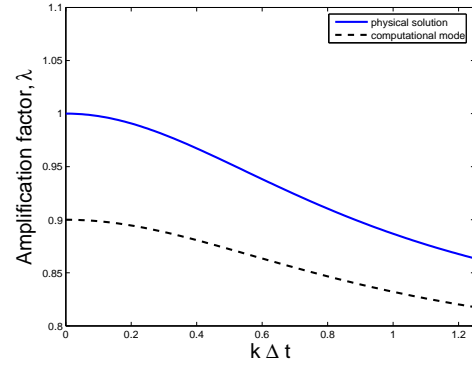


b)

Figure 2. Stability of the semi-implicit leapfrog time integrator with no time-filter. Figure a) uses a value of  $v = .5$  and Figure b) uses a value of  $v = .6$ .



a)



b)

Figure 3. Stability of the semi-implicit leapfrog time integrator with a time filter weight of  $\epsilon = .05$ . Figure a) uses a value of  $v = .5$  and Figure b) uses a value of  $v = .6$ .

### C. HYBRID EULERIAN-LAGRANGIAN SEMI-IMPLICIT TIME-INTEGRATORS

Here, the HELSI method of Giraldo is presented [8]. For this method, the governing equations are rewritten in the Lagrangian form

$$\begin{aligned}\frac{d\pi}{dt} &= -\frac{R}{c_v}\pi\nabla\cdot\mathbf{u} \\ \frac{d\mathbf{u}}{dt} &= -c_p\theta\nabla\pi - g\mathbf{k} + \mu\nabla^2\mathbf{u} \\ \frac{d\theta}{dt} &= \mu\nabla^2\theta.\end{aligned}\tag{3.32}$$

More simply, this can be written as

$$\frac{d\mathbf{q}}{dt} = \mathbf{S}_L(\mathbf{q})\tag{3.33}$$

where  $\mathbf{S}_L(\mathbf{q})$  contains the source terms without advection.

The HELSI discretization is

$$\frac{d\mathbf{q}}{dt} = \{\mathbf{S}_L(\mathbf{q}) - \delta\mathbf{L}(\mathbf{q})\} + \delta[\mathbf{L}(\mathbf{q})].\tag{3.34}$$

As before, the terms inside the braces are time-integrated explicitly while the terms inside the square brackets are time-integrated implicitly. The linearized operator  $\mathbf{L}$  is defined in the same way as before. Applying the HELSI operator yields

$$\mathbf{q}^{n+1} = \sum_{m=0}^1 \alpha_m \tilde{\mathbf{q}}^{n-m} + \gamma\Delta t \sum_{m=0}^2 \beta_m \mathbf{S}_L(\mathbf{q})^{n-m} + \delta\gamma\Delta t \mathbf{L}(\sum_{m=-1}^2 \rho_m \mathbf{q}^{n-m})\tag{3.35}$$

where the tilde above  $\mathbf{q}$  denotes the quantity is determined along characteristics in a semi-Lagrangian sense. The term Hybrid Eulerian-Lagrangian comes from the fact that  $\tilde{\mathbf{q}}$  terms, which correspond to the time derivatives, are computed in the Lagrangian sense, while the remaining terms are computed in the Eulerian sense. The next step is to determine how to compute  $\tilde{\mathbf{q}}$ , the solution values along the characteristics.

The goal is to solve the Lagrangian derivative

$$\frac{d\mathbf{q}}{dt} = 0. \quad (3.36)$$

For this project, the Operator-Integration-Factor Splitting (OIFS) method [9] is used.

## 1. Operator-Integration-Factor Splitting Method

In the OIFS method, Eulerian substepping is used to compute the Lagrangian derivative. The goal is to solve

$$\frac{\partial \mathbf{q}}{\partial s} = -\check{\mathbf{u}} \cdot \nabla \mathbf{q} \quad (3.37)$$

where  $\partial s = \frac{\partial t}{ns}$ ,  $ns$  is the number of substeps, and with  $\mathbf{q}(s=0) = \mathbf{q}^n$  to approximate  $\tilde{\mathbf{q}}^n$  and  $\mathbf{q}(s=1) = \mathbf{q}^{n+1}$  to approximate  $\tilde{\mathbf{q}}^{n+1}$ . To illustrate this procedure, assume RK2 is used to solve equation 3.37. This yields

$$\begin{aligned} \hat{\mathbf{q}}_1^{n+\frac{1}{2}} &= \mathbf{q}^n - \frac{\Delta t}{2} \check{\mathbf{u}}^n \cdot \nabla \mathbf{q}^n \\ \hat{\mathbf{q}}_1^{n+1} &= \mathbf{q}^n - \Delta t \check{\mathbf{u}}^{n+\frac{1}{2}} \cdot \nabla \hat{\mathbf{q}}_1^{n+\frac{1}{2}} \end{aligned} \quad (3.38)$$

and

$$\begin{aligned} \hat{\mathbf{q}}_2^{n-\frac{1}{2}} &= \mathbf{q}^{n-1} - \frac{\Delta t}{2} \check{\mathbf{u}}^{n-1} \cdot \nabla \mathbf{q}^{n-1} \\ \hat{\mathbf{q}}_2^n &= \mathbf{q}^{n-1} - \Delta t \check{\mathbf{u}}^{n-\frac{1}{2}} \cdot \nabla \hat{\mathbf{q}}_2^{n-\frac{1}{2}} \end{aligned} \quad (3.39)$$

$$\begin{aligned} \hat{\mathbf{q}}_2^{n+\frac{1}{2}} &= \hat{\mathbf{q}}_2^n - \frac{\Delta t}{2} \check{\mathbf{u}}^n \cdot \nabla \hat{\mathbf{q}}_2^n \\ \hat{\mathbf{q}}_2^{n+1} &= \hat{\mathbf{q}}_2^n - \Delta t \check{\mathbf{u}}^{n+\frac{1}{2}} \cdot \nabla \hat{\mathbf{q}}_2^{n+\frac{1}{2}} \end{aligned} \quad (3.40)$$

where  $\tilde{\mathbf{q}}^n = \hat{\mathbf{q}}_1^{n+1}$  and  $\tilde{\mathbf{q}}^{n-1} = \hat{\mathbf{q}}_2^{n+1}$  and the velocity field is defined as  $\check{\mathbf{u}}^{n+\frac{1}{2}} = \mathbf{u}(x(t^{n+1}), t^{n+\frac{1}{2}})$ .

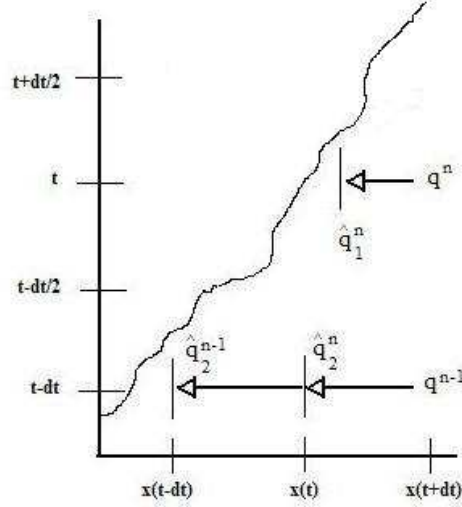


Figure 4. A schematic of the OIFS trajectory computation. The curved line is the actual trajectory, while the arrows denote the paths followed to compute the departure points using an RK2 approximation. Each arrow represents a two-stage process.

A further increase in the maximum allowable timestep can be accomplished by using higher order RK methods, or by substepping. By substepping, each timestep is broken up into smaller timesteps, which are passed to the RK method.

## IV. TEST CASES

### A. CASE 1: RISING THERMAL BUBBLE

The rising thermal bubble is presented in Giraldo and Restelli [2], and is similar to the test proposed by Robert [10]. The initial atmospheric state has no flow, a constant potential temperature field, and is in hydrostatic balance defined by

$$c_p \bar{\theta} \frac{\partial \bar{\pi}}{\partial z} = -g$$

which then yields

$$\bar{\pi} = 1 - \frac{g}{c_p \bar{\theta}} z$$

for the Exner pressure. The following potential temperature perturbation is then added to drive the motion of the air

$$\theta' = \begin{cases} 0 & \text{for } r > r_c \\ \frac{\theta_c}{2} \left[ 1 + \cos \left( \frac{\pi_c r}{r_c} \right) \right] & \text{for } r \leq r_c \end{cases}$$

where  $\theta_c = \frac{1}{2}$  Kelvin,  $\pi_c$  is the trigonometric constant,  $r = \sqrt{(x - x_c)^2 + (z - z_c)^2}$  with the following constants:  $\bar{\theta} = 300$  Kelvin,  $r_c = 250$  meters, and  $(x, z) \in [0, 1000]^2$  meters with  $t \in [0, 600]$  seconds and  $(x_c, z_c) = (500, 350)$  meters. The boundary conditions for all four boundaries are no-flux. The tests are run on a 10 element x 10 element grid with 10 polynomials in each element. Viscosity is ignored. For both the explicit and semi-implicit leapfrog methods, a time-filter weight of .05 is used.

### B. CASE 2: LINEAR HYDROSTATIC MOUNTAIN WAVE

The linear hydrostatic mountain waves is originally presented in Durran and Klemp [11] and Smith [12]. The atmosphere is isothermal, meaning the temperature is constant at  $\bar{T} = 250$  Kelvin. Initially, the atmosphere has a mean flow of  $\bar{u} = 20$  meters/second. Imposing hydrostatic balance requires

$$c_p \bar{\theta} \frac{\partial \bar{\pi}}{\partial z} = -g$$

where  $\bar{\theta} = \frac{\bar{T}}{\bar{\pi}}$

This gives

$$\bar{\pi} = e^{-\frac{g}{c_p T} z}$$

for the Exner pressure.

The simulation is done on the domain  $(x, z) \in [0, 240000] \times [0, 30000]$  meters and is run for 1 hour. The width of the Agnesi mountain profile is

$$h(x, z) = \frac{h_c}{1 + \left(\frac{x-x_c}{a_c}\right)^2} \quad (4.1)$$

where  $h_c = 1$  meter,  $x_c = 120,000$  meters, and  $a_c = 10,000$  meters. The Brunt-Väisälä frequency  $N = \frac{g}{\sqrt{c_p T}}$ . Taking  $N = 0.0195/\text{second}$  yields  $\frac{Na_c}{u} > 1$  which is in the hydrostatic range. The bottom boundary condition is no-flux, while the remaining boundaries use a non-reflecting absorbing boundary condition, which works like a sponge. A spatial resolution of 20 elements in the x-direction, 10 elements in the z-direction, and 10 polynomials per element is used. Viscosity is ignored. For both the explicit and semi-implicit leapfrog methods, a time-filter weight of .05 is used.

### C. CASE 3: DENSITY CURRENT

The density current is proposed in Straka et al. [13]. This case is not too dissimilar to the rising thermal bubble, but there are differences in the size of the domain and the shape of the cold cosine bubble. The potential temperature perturbation is

$$\theta' = \frac{1}{2}\theta_c [1 + \cos(\pi_c r)]$$

where  $\theta_c = -15$  Kelvin,  $\pi_c$  is the trigonometric constant, and

$$r = \sqrt{\left(\frac{x-x_c}{x_r}\right)^2 + \left(\frac{z-z_c}{z_r}\right)^2}.$$

The simulation is done on the domain  $(x, z) \in [0, 25600] \times [0, 6400]$  meters and is run for 900 seconds. As in Straka et al. [13], only half of the horizontal domain

is defined. A no-flux boundary condition is implemented at all four boundaries, and a dynamic viscosity of  $\mu = 75$  meters<sup>2</sup>/second is used. A spatial resolution of 32 elements in the x-direction, 8 elements in the z-direction, and 8 polynomials per element is used. For the explicit leapfrog method, a time-filter weight of .05 is used. For the semi-implicit leapfrog method, a time-filter weight of 0.1 is used.

## D. CASE 4: INERTIA-GRAVITY WAVE

This nonhydrostatic inertia-gravity wave is identical to the test case proposed by Skamarock and Klemp [14]. Initially, the atmosphere is uniformly stratified, has a constant mean flow of  $\bar{u} = 10$  meters/second and a Brunt-Väisälä frequency of  $N = 0.01$ /second. By definition

$$N^2 = g \frac{\partial}{\partial z} (\ln \bar{\theta})$$

yields

$$\bar{\theta} = \theta_0 e^{\frac{N^2}{g} z}$$

where  $\theta_0 = 300$ . Hydrostatic balance is imposed, which yields

$$\bar{\pi} = 1 + \frac{g^2}{c_p \theta_0 N^2} \left( e^{-\frac{N^2}{g} z} - 1 \right).$$

Finally, the potential temperature is perturbed by

$$\theta' = \theta_c \frac{\sin\left(\frac{\pi_c z}{h_c}\right)}{1 + \left(\frac{x - x_c}{a_c}\right)^2}$$

where  $\theta_c = 0.01$  Kelvin,  $h_c = 10,000$  meters,  $a_c = 5,000$  meters, and  $\pi_c = 3.14159265$  is the trigonometric constant. The simulation is done on the domain  $(x, z) \in [0, 300000] \times [0, 10000]$  and is run for 3000 seconds. No-flux boundary conditions are implemented for the top and bottom boundaries and periodic boundary conditions are used for the lateral boundaries. A spatial resolution of 60-elements in the x direction, 2-elements in the z direction, and 10 polynomials per element is used. Viscosity is ignored. For both the explicit and semi-implicit leapfrog methods, a time-filter weight of .01 is used.

THIS PAGE INTENTIONALLY LEFT BLANK



## V. RESULTS

### A. OVERVIEW

To evaluate performance and accuracy, the four test cases were run using each of the time-integration methods. Cases 1 and 4 were run on a single processor, single user computer, while the remaining cases were run on a dual-processor, shared computer. The maximum usable timestep, Courant number (C), and total wallclock time of each of the simulations was recorded. For each test case, specific metrics were chosen to evaluate accuracy. Finally, the results of each simulation were plotted and visually compared as another method of comparison between the time-integration methods.

For the HELSI methods, the OISF method was performed using second (RK2), third (RK3), and fourth (RK4) order Runge-Kutta methods, each with 1, 2, 3, and 4 substeps, for a total of 12 HELSI runs. The run with the shortest wallclock time and the run with the largest Courant number are used for the comparison. These runs may or may not be different, giving one or two “best” HELSI simulations for each case. When multiple HELSI runs have the same Courant number, the one with the shortest total wallclock time is used. The results of all runs are given in Appendix C.

### B. DETERMINING THE MAXIMUM USABLE TIME STEP

For each of the time-integration methods, the maximum timestep was determined experimentally. Using linear stability theory as a first estimate for the explicit methods, an initial guess was made. If the method proved stable at that timestep, it was increased. Otherwise it was decreased. The process was repeated with an increasingly small change until the limit of stability was determined to a high degree of accuracy. Finally, the output of the simulation was plotted to ensure the results were consistent with what was expected.

For the semi-implicit and HELSI methods, a similar approach was used. However, these methods have a tendency to preserve stability, but not accuracy, for increasingly large timesteps. Therefore, after each simulation, the output was plotted, and the numerical results were compared to that of the explicit time-integrators. Using these comparisons, it was clear when the maximum usable timestep was exceeded.

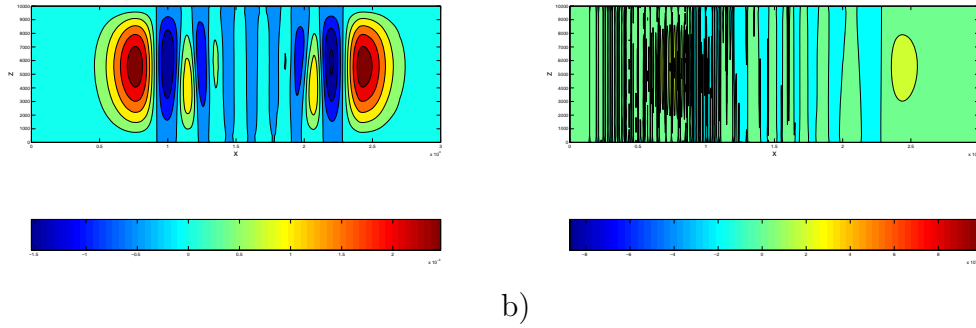


Figure 5. An example of the HELSI time-integrator retaining stability but not accuracy. In figure a), case 4 is time-integrated with a timestep of 4.75 seconds using the HELSI method, a RK2 OIFS method, and one substep. In figure b) case 4 is time integrated using a timestep of 5.75 seconds using the same HELSI method. While the integration remains stable, the result is obviously unusable.

The benefit of an increased time step goes beyond decreasing the wallclock time of the numerical integration. In an operational NWP model, physical parametrizations are done after each time step. These operations can account for as much as 70% of the total computational cost of the forecast integration component of the model run. By increasing the timestep, the total number of physical parametrization calls decreases.

For example, if an advanced numerical time-integration method can double the maximum usable timestep, it will halve the number of computations done running the parametrizations. Even if this method requires the same or slightly more wallclock

time to do the time-integration of the dynamics, the overall effect will be an increase in efficiency due to the reduction of computations used in parametrizations.

## C. CASE 1: RISING THERMAL BUBBLE

### 1. Courant Number and Computational Cost

#### *a. Explicit Methods*

Table I. Case 1: Courant number and computational cost of explicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2	.0018	0.1340	6617
LF2	.0052	0.3871	2387
RK3	.0097	0.7220	2263
RK34	.0120	0.8932	2315
RK35	.0148	1.1020	2102

Of the explicit methods, the SSP RK methods offer the highest degree of efficiency while BDF2 underperforms significantly. Still, the Courant number reaches a limit around 1.

#### *b. Semi-Implicit Methods*

Table II. Case 1: Courant number and computational cost of semi-implicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2 SI	.55	40.94	644.6
LF2 SI	.85	63.27	926.6

As expected, the Semi-Implicit methods offer significant improvement in efficiency over all of the explicit methods. LF2 SI, for example, allows for a Courant number over 50 times greater than the best explicit methods. Meanwhile, BDF2 SI reduces Wallclock time by about 70%.

*c. HELSI Method*

Table III. Case 1: Courant number and computational cost of HELSI method.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK4	4	11.25	837.4	215.60

For the HELSI comparison, depart method refers to the time-integrator used in the OIFS method, while NS is the number of substeps. The HELSI method offers a great improvement in efficiency over the semi-implicit methods, with wallclock time reduced by a factor of about 13 to 20, and offers an even greater improvement over the explicit methods. In fact, HELSI completes the simulation in roughly one tenth the time of RK35, and with a Courant number around 760 times larger.

## 2. Accuracy

Since no analytic solution is available for this test case, the maximum and minimum values of the potential temperature perturbations  $\theta'$  are compared to ensure consistency among the methods. Potential temperature is chosen as this is a thermal problem. Recall that the maximum temperature perturbation within the rising bubble at the start of the simulation is .5 degrees Kelvin. The explicit methods are compared to one another, and the overall most efficient method is chosen for comparison with the advanced methods.

*a. Explicit Methods*

Table IV. Case 1: A comparison of potential temperature perturbations for the explicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2	.513	-.0849
LF2	.534	-.0959
RK3	.533	-.0959
RK34	.535	-.0960
RK35	.535	-.0959

Due to its extreme computational cost, BDF2 is eliminated from the comparison immediately. Amongst the other four methods, the results are virtually identical. Therefore, RK35 is chosen for comparison to the advanced methods because of its efficiency.

***b. Semi-Implicit Methods***

Table V. Case 1: A comparison of potential temperature perturbations for the semi-implicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2	.509	-.0799
LF2	.529	-.0519

Both of the semi-implicit methods give results in line with expectations. BDF2 SI is chosen for comparison as it has a shorter wallclock time than LF2 SI

***c. HELSI Method***

Table VI. Case 1: Potential temperature perturbations for the HELSI time-integrator.

Depart Method	NS	$\theta'_{max}$	$\theta'_{min}$
RK4	4	.521	-.0911

The results of the HELSI simulation are in line with those of the explicit runs, which are known to be very accurate. The best methods are plotted to show that the qualitative results are similar for each.

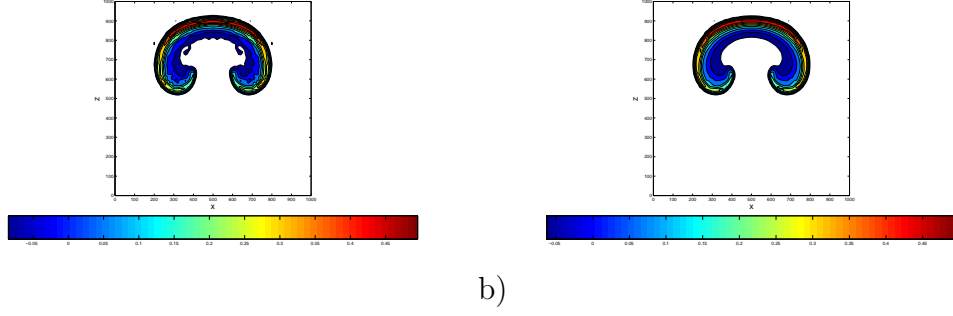


Figure 6. Case1: A comparison of potential temperature perturbations using a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators for case 1 at 3000 sec.

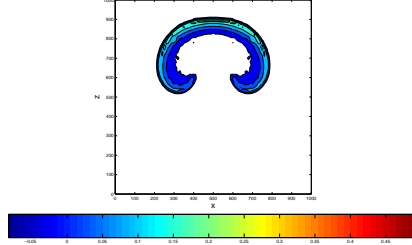


Figure 7. Case 1: A view of potential temperature perturbations using the HELSI time-integrator with RK4 depart method and 4 substeps at 3000 sec.

### 3. Comparison and Conclusions

This is an excellent example of the potential of the HELSI method. Not only does the HELSI method allow for a dramatically increased Courant number and run significantly faster than the semi-implicit methods, it also gives results very similar to the explicit methods, which are known to be very accurate.

Figure 6 b) shows that the BDF2 SI method seems to smooth the underside of the thermal bubble, while the HELSI method in Figure 7 seems to retain more of the rigid structure of the RK35 method. This increase in accuracy over the semi-implicit method, coupled with the dramatic increase in efficiency, makes HELSI the logical choice for case 1.

## D. CASE 2: LINEAR HYDROSTATIC MOUNTAIN WAVE

### 1. Courant Number and Computational Cost

#### *a. Explicit Methods*

Table VII. Case 2: Courant number and computational cost of explicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2	.07	.0584	2184.37
LF2	.18	.150	866.24
RK3	.34	.284	784.92
RK34	.43	.359	746.26
RK35	.53	.442	714.10

Again, the SSP RK3 methods are the most efficient of the explicit time-integrators, and BDF2 is, by far, the slowest.

#### *b. Semi-Implicit Methods*

Table VIII. Case 2: Courant number and computational cost of semi-implicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2 SI	6.0	5.008	287.59
LF2 SI	11.75	9.807	295.42

The semi-implicit methods also behave similarly to the first test case. Here, LF2 SI increases the Courant number more than 20 fold, while both methods run the simulation in less than half the time of RK35, the most efficient explicit method.

*c. HELSI Methods*

Table IX. Case 2: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	13.0	10.85	231.87
RK3	1	17.0	14.19	250.74

Again, the HELSI method yields great results. However, there appears to be a limitation on the Courant number which is not directly related to the OIFS method. Increasing the order of the OIFS method and number of substeps cannot increase the Courant number beyond 14.19. Therefore, it makes the most sense to use the simplest OIFS method that can achieve this Courant number, as increasing the OIFS method further will only slow the integration without benefit. Still, HELSI once again allows for a significantly higher Courant number and a slightly shorter wallclock time than the semi-implicit methods.

## 2. Accuracy

Unlike the other test cases, an analytical solution is available for the linear hydrostatic mountain wave. This allows for a more robust comparison of the accuracy of each of the time-integration methods. Root mean square errors of each of the four perturbation variables are compared, with smaller values representing better error characteristics.



**a. Explicit Methods**

Table X. Case 2: A comparison of RMS errors for the explicit time-integrators.

Method	$\pi'_{RMS}$	$u'_{RMS}$	$v'_{RMS}$	$\theta'_{RMS}$
BDF2	$4.62 \times 10^{-7}$	$8.33 \times 10^{-3}$	$4.64 \times 10^{-4}$	$6.29 \times 10^{-3}$
LF2	$4.62 \times 10^{-7}$	$8.33 \times 10^{-3}$	$4.64 \times 10^{-4}$	$6.30 \times 10^{-3}$
RK3	$4.62 \times 10^{-7}$	$8.33 \times 10^{-3}$	$4.64 \times 10^{-4}$	$6.29 \times 10^{-3}$
RK34	$4.62 \times 10^{-7}$	$8.33 \times 10^{-3}$	$4.64 \times 10^{-4}$	$6.29 \times 10^{-3}$
RK35	$4.62 \times 10^{-7}$	$8.33 \times 10^{-3}$	$4.64 \times 10^{-4}$	$6.29 \times 10^{-3}$

Once again the explicit methods all yield very similar results. Since all the error characteristics are virtually identical, RK35 is again chosen for comparison, as it is the most efficient method.

**b. Semi-Implicit Methods**

Table XI. Case 2: A comparison of RMS errors for the semi-implicit time-integrators.

Method	$\pi'_{RMS}$	$u'_{RMS}$	$v'_{RMS}$	$\theta'_{RMS}$
BDF2 SI	$4.61 \times 10^{-7}$	$8.31 \times 10^{-3}$	$4.63 \times 10^{-4}$	$6.28 \times 10^{-3}$
LF2 SI	$4.61 \times 10^{-7}$	$8.34 \times 10^{-3}$	$4.65 \times 10^{-4}$	$6.29 \times 10^{-3}$

For this test case, the semi-implicit methods seem to perform about as well as the explicit methods.

**c. HELSI Method**

Table XII. Case 2: A comparison of RMS errors for the HELSI time-integrators.

Depart Method	NS	$\pi'_{RMS}$	$u'_{RMS}$	$v'_{RMS}$	$\theta'_{RMS}$
RK2	1	$4.54 \times 10^{-7}$	$8.27 \times 10^{-3}$	$4.51 \times 10^{-4}$	$6.23 \times 10^{-3}$
RK3	1	$4.56 \times 10^{-7}$	$8.50 \times 10^{-3}$	$4.71 \times 10^{-4}$	$6.26 \times 10^{-3}$

The HELSI simulation using RK2 for the OIFS method actually yields slightly better error characteristics than both the explicit and semi-implicit time-integrators. The simulation using RK3 for the OIFS method, however, introduces slightly larger error characteristics into the velocity vector. Still, these errors are quite good, and, given the overall increase in efficiency, are likely acceptable.

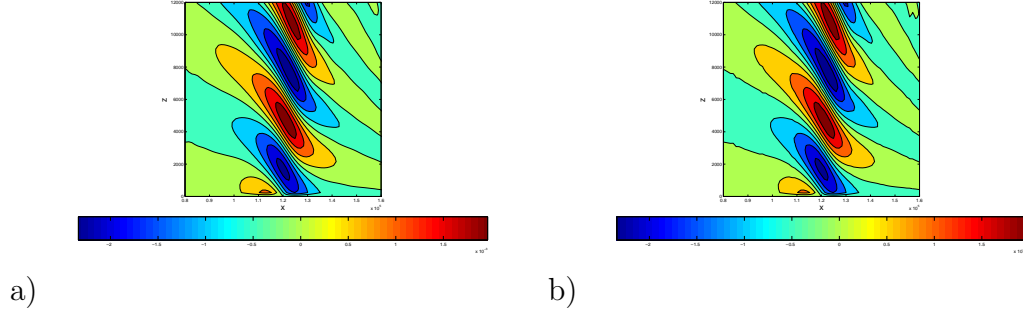


Figure 8. Case 2: A comparison of vertical velocity perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 1 hr.

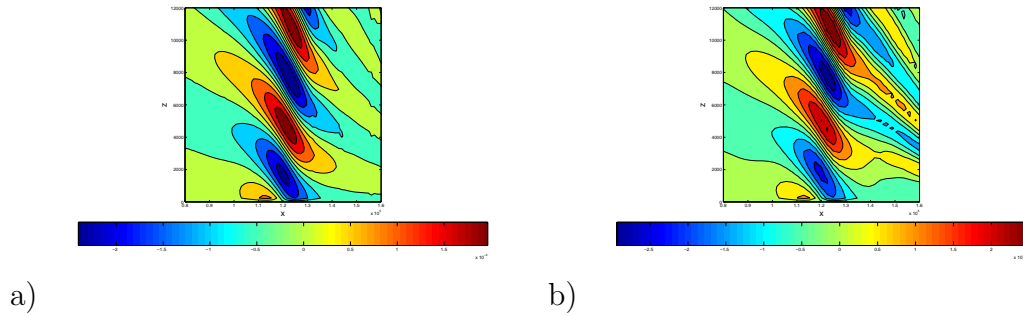


Figure 9. Case 2: A comparison of the vertical velocity perturbations for a) the HELSI method with depart method RK2 and 1 substep and b) the HELSI method with depart method RK3 and 1 substep at 1 hr.

### 3. Comparison and Conclusions

Once again the HELSI time-integrator is, by far, the most efficient time integration method. However, this test case exposes a limitation of the HELSI method.

The Courant number could not be increased beyond 14.19. The likely limiting factor is the linearization used in the semi-implicit part of the HELSI operator. As the timestep becomes increasingly large, the linear approximations break down, and the solution cannot converge.

The accuracy of the HELSI method also seems to suffer slightly for this test case. While the RMS errors of both HELSI simulations are quite close to those of the explicit runs, Figure 9 b), where  $\Delta t = 17$  seconds, shows some artifacts on the right side of the graphic which are not seen in the explicit runs. Figure 9 a), with  $\Delta t = 13$  seconds, is more inline with the RK35 method, and still offers an improvement in both Courant number and wallclock time over the semi-implicit methods.

Overall, the HELSI method again seems to be the best choice. The benefits may not be as dramatic as the first test case, yet HELSI still offers a significant improvement.

## E. CASE 3: DENSITY CURRENT

### 1. Courant Number and Computational Cost

#### *a. Explicit Methods*

Table XIII. Case 3: Courant number and computational cost of explicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2	.02	.1336	1542.40
LF2	.0575	.3841	549.88
RK3	.115	.7628	607.47
RK34	.140	.9352	564.17
RK35	.175	1.1690	565.22

For the density current, BDF2 is, once again, the least efficient time-integrator. LF2 is surprisingly efficient, with a wallclock time slightly lower than

RK35. Still, given the larger Courant number of RK35, it should still be considered the most efficient of the explicit methods.

### *b. Semi-Implicit Methods*

Table XIV. Case 3: Courant number and computational cost of semi-implicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2 SI	.625	4.173	243.17
LF2 SI	.375	2.504	624.67

LF2 SI struggles with this test case. The Courant number is only double that of RK35, while the wallclock time is actually higher than all of the explicit methods except BDF2. This is likely due to the inclusion of viscosity, which is unique to this case, as the leapfrog method is unstable for diffusive problems. BDF2 SI, however, offers significant improvements over the explicit methods.

### *c. HELSI Methods*

Table XV. Case 3: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK3	1	1.2	8.012	320.12

Once again the HELSI methods seem to have a limitation independent of the OIFS method (see appendix C). Still, the Courant number is nearly doubled over BDF2 SI. Wallclock time, however, has increased by nearly 33%.

## **2. Accuracy**

The density current is another case without an analytic solution. Like the rising bubble, it is also a thermal problem. For this reason, potential temperature perturbations are once again the choice for comparison.

*a. Explicit Methods*

Table XVI. Case 3: A comparison of potential temperature perturbations for the explicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2	0.294	-9.32
LF2	0.281	-9.06
RK3	0.279	-9.06
RK34	0.271	-9.02
RK35	0.268	-8.98

For the density current, there is actually a bit of difference in the results of the explicit operators. Since no analytic solution is available, it is impossible to say which of these results is the most accurate. Still, it is expected the truth should lie somewhere within this range.

*b. Semi-Implicit Methods*

Table XVII. Case 3: A comparison of potential temperature perturbations for the semi-implicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2 SI	.289	-9.02
LF2 SI	.236	-9.15

Of the two semi-implicit methods, BDF2 yields results closest to those of the explicit methods. Given the difference in the results produced by LF2 SI, and its poor performance (see table XIV), it does not seem to be a good option for this case.

*c. HELSI Method*

Table XVIII. Case 3: Potential temperature perturbations for the HELSI time-integrator.

Depart Method	NS	$\theta'_{max}$	$\theta'_{min}$
RK3	1	.217	-9.14

HELSI seems to be underestimating  $\theta'_{max}$ . Still, as this is a cold bubble problem,  $\theta'_{min}$  is the most important value to consider. Therefore, the results are still in line with the other time-integrators.

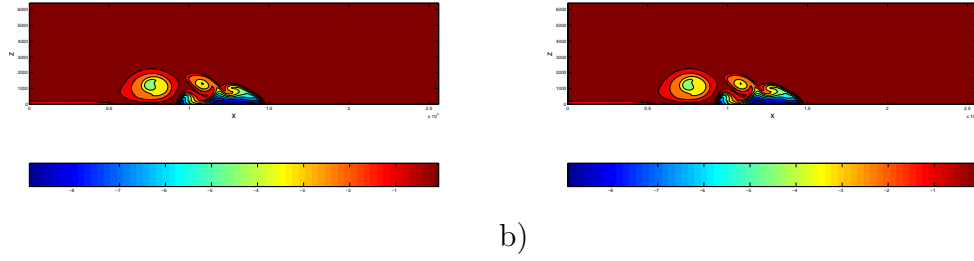


Figure 10. Case 3: A comparison of potential temperature perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 900 sec.

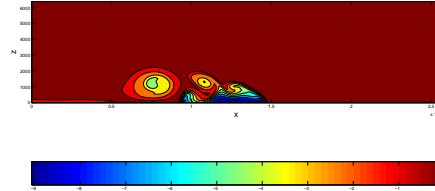


Figure 11. Case 3: Potential temperature perturbations for the HELSI method with depart method RK3 and 1 substep at 900 sec.

### 3. Comparison and Conclusions

Once again, the HELSI method has a limitation other than the OIFS method. For the density current, this limitation may also be due to the linearization, or may be due to the inclusion of viscosity. In the HELSI algorithm, viscous terms are treated explicitly, thus, they have a limited stability region.

The wallclock time for HELSI is actually larger than that of BDF2 SI. Still, HELSI offers a significant increase in the Courant number. This improvement will probably prove worth the increase in wallclock time in an operational model, as parametrizations will only be called about half as often.

Figures 10 and 11 show no obvious differences between the time integration methods. This is consistent with the similar perturbation values in Tables XVI, XVII, and XVIII. Once again, the HELSI method is the best performer overall, but the BDF2 SI is not as far off as in the previous cases.

## F. CASE 4: INERTIAL GRAVITY WAVE

### 1. Courant Number and Computational Cost

#### *a. Explicit Methods*

Table XIX. Case 4: Courant number and computational cost of explicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2	.1	.1573	794.4
LF2	.26	.4089	316.6
RK3	.47	.7392	299.4
RK34	.58	.9122	299.6
RK35	.72	1.1320	298.3

Once again, the explicit time-integrators follow the same performance trend. BDF2 underperforms significantly, while RK35 is the most efficient.

***b. Semi-Implicit Methods***

Table XX. Case 4: Courant number and computational cost of semi-implicit methods.

Method	Max Timestep (s)	C	Wallclock Time (s)
BDF2 SI	2.3	3.617	111.55
LF2 SI	6.3	9.908	168.8

The semi-implicit time-integrators once again offer significant improvements over their explicit counterparts. LF2 SI allows for a Courant number nearly 9 times greater than that of RK35, while BDF2 SI cuts the wallclock time in half.

***c. HELSI Methods***

Table XXI. Case 4: Courant number and computational cost of HELSI methods.

Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	4.75	7.470	141.63
RK2	3	9.25	14.55	258.47

The HELSI approach, once again allows for a significant increase in Courant number. Wallclock time, however, is more than doubled to achieve this increase.

## **2. Accuracy**

For this case, there is no obvious choice of metric to compare accuracy. Therefore, to be consistent, potential temperature perturbations are used.



*a. Explicit Methods*

Table XXII. Case 4: A comparison of potential temperature perturbations for the explicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2	.00279	-.00152
LF2	.00279	-.00152
RK3	.00279	-.00152
RK34	.00279	-.00152
RK35	.00279	-.00152

Since all of the explicit methods give the exact same potential temperature perturbations, RK35 is once again the method used for comparison, as it is the most efficient.

*b. Semi-Implicit Methods*

Table XXIII. Case 4: A comparison of potential temperature perturbations for the semi-implicit time-integrators.

Method	$\theta'_{max}$	$\theta'_{min}$
BDF2 SI	.00280	-.00152
LF2 SI	.00278	-.00147

The semi-implicit time-integrators give very similar results to the explicit ones, and are a lot more efficient.

*c. HELSI Methods*

Table XXIV. Case 4: A comparison of potential temperature perturbations for the HELSI time-integrators.

Depart Method	NS	$\theta'_{max}$	$\theta'_{min}$
RK2	1	.00287	-.00156
RK2	3	.00280	-.00152

The HELSI methods also give very similar results.

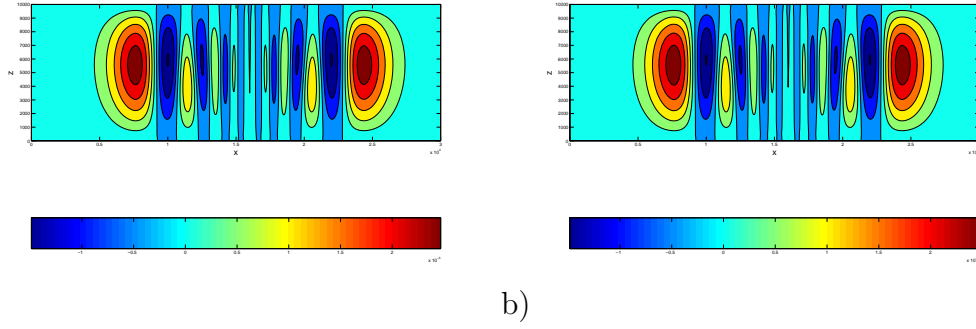


Figure 12. Case 4: A comparison of potential temperature perturbations for a) the RK35 explicit and b) the BDF2 semi-implicit time-integrators at 3000 sec.

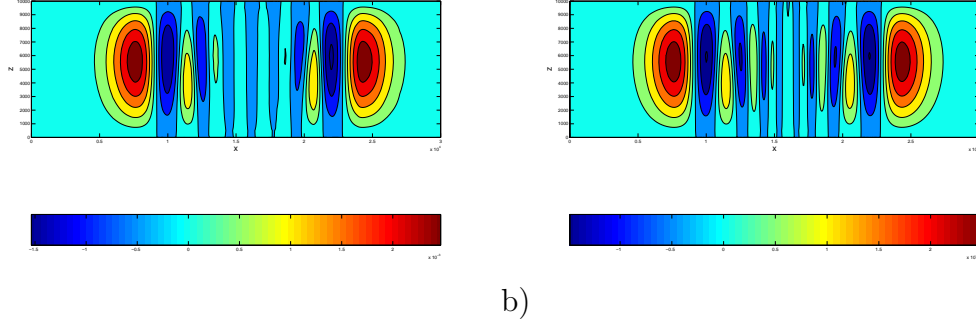


Figure 13. Case 4: A comparison of potential temperature perturbations for the a) HELSI method with depart method RK2 and 1 substep and b) the HELSI method with depart method RK2 and 3 substeps at 300 sec.

### 3. Comparison and Conclusions

The limitation on the HELSI approach caused by the semi-implicit linearization is significant for this case. While the Courant number is improved over the semi-implicit time-integrators, the difference is not nearly as great as in the other cases. Meanwhile, the wallclock times for the HELSI methods are significantly higher than those of the semi-implicit approach. However, in an operational model, the increase in Courant number might still make the HELSI method more efficient overall when parametrizations are included. This is not a foregone conclusion, however, as it is in the other test cases.

Figures 12 and 13 show that all the time-integrators give similar results. Of the HELSI simulations, the RK2 OIFS method with 3 substeps allows for the largest Courant number, and, therefore, would be the method of choice.

Overall, the increased Courant number likely makes HELSI the best option. The semi-implicit approach, however, is not as far behind as the previous cases. Nonetheless, both methods offer dramatic improvements over explicit time-integrators.

THIS PAGE INTENTIONALLY LEFT BLANK

## VI. CONCLUSIONS AND RECOMMENDATIONS

HELSEI time-integrators offer a dramatic improvement in Courant number when compared to semi-implicit and explicit time-integrators. Their accuracy approaches that of the explicit methods, and meets or exceeds that of the semi-implicit time-integrators in most cases. Considering the vast improvements in efficiency, the operational use of HELSEI time-integrators seems to be a logical approach.

Implementing HELSEI time-integrators in an NWP model would free significant computational resources to be used in other areas. They would allow for significant improvements in data assimilation, parametrizations, and/or spatial resolution without large increases in hardware requirements. The overall benefits of these improvements would, in all likelihood, result in a better forecast, even though the HELSEI methods may be slightly less accurate for a given resolution than currently used split-explicit methods.

In order for their potential to be fully realized, however, a few issues need to be addressed. One limitation of the current HELSEI time-integrators is in the handling of viscosity. Currently, viscosity is handled explicitly in the HELSEI algorithm. Treating viscosity in the semi-implicit part of the algorithm may alleviate some of the problems it causes.

Another possible improvement to the HELSEI approach would be the use of the RK35 scheme in the OIFS method. Since this method proved to be the most efficient of the explicit methods tested, it is likely that it would provide the best efficiency to the OIFS approach. This improvement, however, could only be realized if the other limitations were first addressed, as the HELSEI method is currently limited by other factors.

Finally, the large Courant number of the HELSEI operator exposes a weakness in the approach. The linearization used in the semi-implicit part of the HELSEI method

often breaks down with large timesteps. A possible remedy would be to implement a fully-implicit scheme, which is not a trivial task. However, the potential benefits of a fully implicit method make it worth continuing to explore despite the difficulties.

This work has shown that, overall, the HELSI approach has several properties that are superior to those of current methods, even with its shortcomings. Addressing these problems can only make HELSI time-integrators an even greater asset to the operational NWP community.

## APPENDIX A. COEFFICIENTS FOR RK35 METHOD

coefficient	value
$a_1$	0.377268915331368
$a_2$	0.377268915331368
$a_3$	0.355909775063327
$a_4$	0.644090224936674
$a_5$	0.242995220537396
$a_6$	0.367933791638137
$a_7$	0.632066208361863
$a_8$	0.238458932846290
$a_9$	0.762406163401431
$a_{10}$	0.237593836598569
$a_{11}$	0.287632146308408

THIS PAGE INTENTIONALLY LEFT BLANK



# APPENDIX B. STABILITY ANALYSIS OF LEAPFROG METHODS WITH TIME-FILTERING

## 1. EXPLICIT LEAPFROG WITH TIME-FILTER

Starting with equation 3.3 and 3.4, the definition of the time-filtered leapfrog method becomes

$$\mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n-1} + 2\Delta t(S(\mathbf{q}^n)). \quad (\text{B.1})$$

Consider the equation

$$\frac{\partial \mathbf{q}}{\partial t} = ik\mathbf{q} \quad (\text{B.2})$$

with solutions

$$\mathbf{q} = Ae^{ikt} \quad (\text{B.3})$$

for the non-filtered solution and

$$\tilde{\mathbf{q}} = \tilde{A}e^{ikt} \quad (\text{B.4})$$

for the filtered solution.

If the equations are discretized, then, at timestep  $n$ ,  $t = n\Delta t$ ,

$$\mathbf{q}^n = Ae^{ikn\Delta t} = A(e^{ik\Delta t})^n \quad (\text{B.5})$$

$$\tilde{\mathbf{q}}^n = \tilde{A}e^{ikn\Delta t} = \tilde{A}(e^{ik\Delta t})^n \quad (\text{B.6})$$

$$S(\mathbf{q})^n = ik\mathbf{q}^n = ikA(e^{ik\Delta t})^n \quad (\text{B.7})$$

and

$$\mathbf{S}(\tilde{\mathbf{q}})^n = ik\mathbf{q}^n = ik\tilde{A}(e^{ik\Delta t})^n. \quad (\text{B.8})$$

Consider the case where there is a time-filter, then

$$A(e^{ik\Delta t})^{n+1} = \tilde{A}(e^{ik\Delta t})^{n-1} + 2\Delta t ik A(e^{ik\Delta t})^n \quad (\text{B.9})$$

or

$$A\lambda^{n+1} = \tilde{A}\lambda^{n-1} + 2ik\Delta t\lambda^n \quad (\text{B.10})$$

where  $\lambda = e^{ik\Delta t}$ . Dividing by  $\lambda$  yields

$$A\lambda^2 = \tilde{A} + 2ik\Delta t\lambda \quad (\text{B.11})$$

From the definition of the time-filter

$$\tilde{\mathbf{q}}^n = \mathbf{q}^n + \epsilon(\mathbf{q}^{n+1} - 2\mathbf{q}^n + \tilde{\mathbf{q}}^{n-1}) \quad (\text{B.12})$$

it is clear that

$$\tilde{A}\lambda = A\lambda + \epsilon(A\lambda^2 - 2A\lambda + \tilde{A}) \quad (\text{B.13})$$

or

$$(\lambda - \epsilon)\tilde{A} = (A - 2A\epsilon)\lambda + A\epsilon\lambda^2. \quad (\text{B.14})$$

Multiplying Equation B.11 by  $(\lambda - \epsilon)$  yields

$$A(\lambda - \epsilon)\lambda^2 = (\lambda - \epsilon)\tilde{A} + 2ik(\lambda - \epsilon)\Delta t\lambda. \quad (\text{B.15})$$

Using Equation B.14, this can be rewritten as

$$A(\lambda - \epsilon)\lambda^2 = (A - 2A\epsilon)\lambda + A\epsilon\lambda^2 + 2ik(\lambda - \epsilon)\Delta t\lambda \quad (\text{B.16})$$

where  $\lambda = 0$  is one factor and can be removed immediately, as it is trivial. Therefore, dividing by  $A\lambda$  yields

$$\lambda^2 - \epsilon\lambda = 1 - 2\epsilon + \epsilon\lambda + 2ik\Delta t\lambda - 2ik\epsilon\Delta t \quad (\text{B.17})$$

or

$$\lambda^2 + (-2\epsilon - 2ik\Delta t)\lambda + (-1 + 2\epsilon + 2ik\epsilon\Delta t) \quad (\text{B.18})$$

whose two solutions are easily found using the quadratic equation. The magnitudes are then plotted to give Figure 1.

## 2. SEMI-IMPLICIT LEAPFROG WITH TIME-FILTER

Starting with Equation 3.28

$$\mathbf{q}^{n+1} = \mathbf{q}^{n-1} + 2\Delta t[\mathbf{L}(v\mathbf{q}^{n+1}) + \mathbf{L}((1-v)\mathbf{q}^{n-1})] \quad (\text{B.19})$$

adding the time-filter yields

$$\mathbf{q}^{n+1} = \tilde{\mathbf{q}}^{n-1} + 2\Delta t[\mathbf{L}(v\mathbf{q}^{n+1}) + \mathbf{L}((1-v)\tilde{\mathbf{q}}^{n-1})]. \quad (\text{B.20})$$

Using Equations B.5-B.8 and the substitution  $\lambda = e^{ik\Delta t}$ , the time-filtered LF2 SI method becomes

$$A\lambda^2 = \tilde{A} + 2\Delta t[Avik\lambda^2 + \tilde{A}(1-v)ik]. \quad (\text{B.21})$$

Multiplying by  $(\lambda - \epsilon)$  yields

$$A\lambda^2(\lambda - \epsilon) = \tilde{A}(\lambda - \epsilon) + 2\Delta t[A(\lambda - \epsilon)vik\lambda^2 + \tilde{A}(\lambda - \epsilon)(1-v)ik]. \quad (\text{B.22})$$

Substituting in Equation B.14, factoring out  $A\lambda$ , and simplifying yields

$$(1 - 2ik\Delta tv)\lambda^2 + (-2\epsilon + 4\epsilon ik\Delta tv - 2\epsilon ik\Delta t)\lambda + (-1 + 2\epsilon + 2(1 - 2\epsilon)ik\Delta t(1 - v)) \quad (\text{B.23})$$

whose two solutions can be found using the quadratic equation. Plotting their magnitude gives Figure 3.

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX C. COMPLETE RESULTS FOR HELSI METHODS

Table XXV. Case 1: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	1	74.44	1440.8
	2	1.8	134	930.94
	3	2.8	208.4	719.14
	4	3.6	268.0	595.78
RK3	1	2.25	167.5	828.55
	2	4.00	297.7	499.28
	3	6.00	446.6	399.15
	4	7.75	576.9	352.39
RK4	1	3.0	223.3	625.37
	2	5.75	428	384.56
	3	8.75	651.3	269.83
	4	11.25	837.4	215.60

Table XXVI. Case 2: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	13.0	10.85	231.87
	2	17.0	14.19	256.98
	3	17.0	14.19	267.48
	4	17.0	14.19	292.03
RK3	1	17.0	14.19	250.74
	2	17.0	14.19	277.77
	3	17.0	14.19	305.28
	4	17.0	14.19	328.70
RK4	1	17.0	14.19	259.56
	2	17.0	14.19	295.74
	3	17.0	14.19	333.78
	4	17.0	14.19	370.11

Table XXVII. Case 3: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	0.9	6.01	325.36
	2	1.15	7.679	371.15
	3	1.2	8.012	398.29
	4	1.2	8.012	449.39
RK3	1	1.2	8.012	320.12
	2	1.2	8.012	457.64
	3	1.2	8.012	487.61
	4	1.2	8.012	637.44
RK4	1	1.2	8.012	346.61
	3	1.2	8.012	479.58
	3	1.2	8.012	589.83
	4	1.2	8.012	701.58

Table XXVIII. Case 4: Courant number and computational cost of HELSI methods.

Depart Method	NS	Max Timestep (s)	C	Wallclock Time (s)
RK2	1	4.75	7.470	141.63
	2	9	14.15	257.16
	3	9.25	14.55	258.47
	4	9.25	14.55	291.22
RK3	1	9	14.15	251.07
	2	9.25	14.15	289.94
	3	9.25	14.15	326.63
	4	9.25	14.15	357.31
RK4	1	9.25	14.15	271.13
	3	9.25	14.15	319.39
	3	9.25	14.15	364.80
	4	9.25	14.15	410.88

# LIST OF REFERENCES

- [1] F.X. Giraldo. Semi-implicit time integrators for a scalable spectral element atmospheric model. *Q. J. R. Meteorol. Soc.*, 131:2431–2454, 2005.
- [2] F.X. Giraldo and M. Restelli. A study of spectral element and discontinuous galerkin methods for mesoscale atmospheric modeling: Equation sets and test cases. *Journal of Computational Physics*, in review, 2007.
- [3] J. Dea. Personal Communication.
- [4] F.X. Giraldo. Time-integrators. Lecture Notes.
- [5] B. Cockburn and C-W Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws ii: General framework. *Mathematics of Computation*, 52:411–435, 1989.
- [6] S.J. Ruuth and R.J. Spiteri. High-order strongly-stability-preserving runge-kutta methods with downwind-biased spatial discretizations. *SIAM Journal on Numerical Analysis*, 42:974–996, 2004.
- [7] D. Durran. *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. Springer, New York, 1999.
- [8] F.X. Giraldo. Hybrid eulerian-lagrangian semi-implicit time-integrators. *Computers and Mathematics with Applications*, 52:1325–1342, 2006.
- [9] Y. Maday, A.T. Patera, and E.M. Ronquist. An operator-integration-factor splitting method for time dependent problems: Application to incompressible fluid flow. *Journal of Scientific Computing*, 5:263–292, 1990.
- [10] A. Robert. Bubble convection experiments with a semi-implicit formulation of the euler equations. *Journal of the Atmospheric Sciences*, 50:1865–1873, 1993.
- [11] D.R. Durran and J.B. Klemp. A compressible model for the simulation of moist mountain waves. *Monthly Weather Review*, 111:2341–2361, 1983.
- [12] R.B. Smith. The influence of mountains on the atmosphere. *Advances in Geophysics*, 21:87, 1979.
- [13] J.M. Straka, R.B. Wilhelmson, L.J. Wicker, J.R. Anderson, and K.K. Droegemeier. Numerical solutions of a non-linear density current: a benchmark solution and comparisons. *International Journal for Numerical Methods in Fluids*, 17:1–22, 1993.

- [14] W.C. Skamarock and J.B. Klemp. Efficiency and accuracy of the klemp-wilhelmson time-splitting technique. *Monthly Weather Review*, 122:2623–2630, 1994.



## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Air Force Institute of Technology  
Wright-Patterson Air Force Base, Ohio
4. Professor Philip Durkee  
Naval Postgraduate School  
Monterey, California
5. Professor Clyde Scandrett  
Naval Postgraduate School  
Monterey, California
6. Professor Francis Giraldo  
Naval Postgraduate School  
Monterey, California
7. CDR Rebecca Stone  
Naval Postgraduate School  
Monterey, California
8. Lt. Col. Karl Pfeiffer  
Naval Postgraduate School  
Monterey, California
9. Maj. Tony Eckel  
Naval Postgraduate School  
Monterey, California
10. CDR Denise Kruse  
Naval Postgraduate School  
Monterey, California